applied sciences

MDPI

Article

Paraphrase Identification with Lexical, Syntactic and Sentential Encodings

Sheng Xu [1,2], Xingfa Shen [1], Fumiyo Fukumoto [3,*], Jiyi Li [3], Yoshimi Suzuki [3] and Hiromitsu Nishizaki [3]

1 School of Computer Science and Technology, Hangzhou Dianzi University, HangZhou 310018, China ; 181050042@hdu.edu.cn (S.X.); shenxf@hdu.edu.cn (X.S.)
2 Integrated Graduate School of Medicine, Engineering, and Agricultural Sciences, Faculty of Engineering, University of Yamanashi, Kofu 400-8511, Japan
3 Graduate Faculty of Interdisciplinary Research, University of Yamanashi, Kofu 400-8511, Japan; jyli@yamanashi.ac.jp (J.L.); ysuzuki@yamanashi.ac.jp (Y.S.); hnishi@yamanashi.ac.jp (H.N.)
* Correspondence: fukumoto@yamanashi.ac.jp; Tel.: +81-55-220-8509

Received: 30 April 2020; Accepted: 9 June 2020; Published: 16 June 2020

Abstract: Paraphrase identification has been one of the major topics in Natural Language Processing (NLP). However, how to interpret a diversity of contexts such as lexical and semantic information within a sentence as relevant features is still an open problem. This paper addresses the problem and presents an approach for leveraging contextual features with a neural-based learning model. Our Lexical, Syntactic, and Sentential Encodings (LSSE) learning model incorporates Relational Graph Convolutional Networks (R-GCNs) to make use of different features from local contexts, i.e., word encoding, position encoding, and full dependency structures. By utilizing the hidden states obtained by the R-GCNs as well as lexical and sentential encodings by Bidirectional Encoder Representations from Transformers (BERT), our model learns the contextual similarity between sentences effectively. The experimental results by using the two benchmark datasets, Microsoft Research Paraphrase Corpus (MRPC) and Quora Question Pairs (QQP) show that the improvement compared with the baseline, BERT sentential encodings model, was 1.7% F1-score on MRPC and 1.0% F1-score on QQP. Moreover, we verified that the combination of position encoding and syntactic features contributes to performance improvement.

Keywords: paraphrase identification; encodings, R-GCNs; BERT; contextual features

1. Introduction

Paraphrase identification is the task to identify whether a pair of sentences is a paraphrase or not. It is highly related to the task of semantic textual similarity to measure the degree of semantic equivalence between two sentences and has been an interest as it is necessary to accomplish most NLP tasks such as question answering, information retrieval, textual entailment, and text summarization. With a recent surge of interest in neural networks, paraphrase identification based on deep learning techniques has been intensively studied. These attempts include Convolutional Neural Networks (CNNs) based model [1,2], Long Short-Term Memory (LSTM) [3], Bidirectional-LSTM (BiLSTM) [4], and gated recurrent averaging [5]. It enables us to utilize the contexts of the target sentences which are powerful for learning features from the training data. Despite some successes, the approaches explored so far rely on word sequence, not making use of different aspects of contexts simultaneously. Several efforts have been made to utilize different representations of the contexts. One attempt is pre-trained contextualized word/sentence representations [5–11]. They have been successfully applied to many NLP tasks, while they explicitly rely on not syntax but the sequential context of words by utilizing a large volume of data.

Motivated by the previous work mentioned in the above, we incorporate several contextual features into a unified framework, Relational Graph Convolutional Networks (R-GCNs) [12–14]. Consider the two sentences from the MRPC data shown in Figure 1. These two sentences are an example of non-paraphrase sentence pair. Adjacent words such as "Hong" "Kong" and "South" "Korea" marked with blue indicate compound nouns and those marked with red such as "0.2–0.4" "percent" and "0.3" "percent" show numeric modifiers. These sentences have different contents/meanings, while there exist many overlapping words such as "Australia", "Singapore", "flat" and "percent". The relative position information marked with blue and red is good indicators to discriminate whether these sentences are a paraphrase or not. Similarly, in the top sentence, "Korea" modifies "lost" with the *nsubj* (*nominal subject*) relation type, while in the second sentence, "Korea" modifies "added" with the *nsubj* relation type. This syntactic structure information also becomes clues that these sentences are not a paraphrase.

Our R-GCNs model integrates different features: (i) word encoding; (ii) position encoding; and (iii) full dependency structures as syntactic encoding from a sentence. We used word encoding obtained by Bidirectional Encoder Representations from Transformers (BERT) [11]. BERT models were pre-trained using a large corpus of sentences. The training is done by masking a few tokens in a sentence and the task is to predict the masked tokens. It learns to produce a powerful internal representation of words as word embedding. Position encoding is a technique to inject information about a token's position within a sentence into a deep learning model. We applied the Stanford parser [15,16] to the input sentences and obtained full dependency structures. Besides contexts with syntactic level, our Lexical, Syntactic, and Sentential Encodings (LSSE) learning model also makes use of contextual information with lexical and sentential levels obtained by the BERT model. Intuitively, by sharing rich contextual features, the model can produce a more meaningful representation to identify paraphrases.

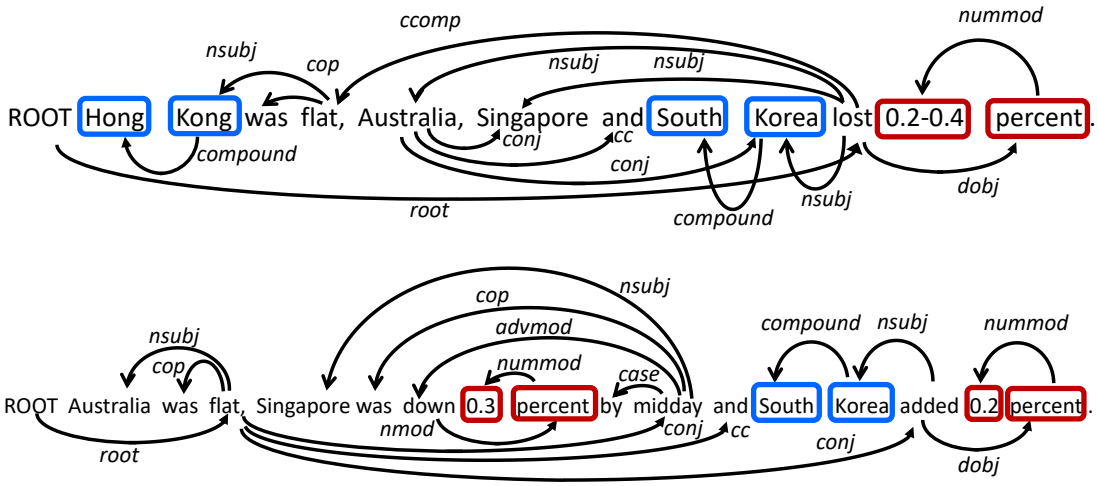

Figure 1. Non-paraphrase sentence pair from the MRPC corpus: Adjacent words such as "Hong" and "Kong" marked with blue indicate compound nouns and those such as "0.2–0.4" and "percent" marked with red show numeric modifiers.

The main contributions of our work can be summarized: (1) We propose a paraphrase identification method that makes use of contextual information with lexical, syntactic, and sentential levels; (2) We apply R-GCNs to utilize different features from local contexts; (3) The experimental results on the two benchmark datasets show that our model is comparable to the related work, and especially, the combination of syntactic features and position encoding contributes to performance improvement in our method.

2. Related Work

There is a large body of work on paragraph identification based on deep learning techniques. The early attempts include a recursive neural network (RNN) [17], CNNs [1,18], and a tree-based LSTM [19]. Despite some successes, techniques explored so far rely on word sequence, ignoring to make use of different aspects of contexts simultaneously.

Several efforts have been made to handle different representations for the same sentence in different contexts. One attempt is pre-trained contextualized language representations. Many authors have attempted to learn contextualized language representation by pre-training a language model with a large amount of unannotated data [7,9,20,21]. Melamud et al. proposed a method called context2vec which learns each sense annotation in the training data by using a bidirectional LSTM trained on an unlabeled corpus [7]. Peters et al. attempted to learn a model called Embeddings from Language Models (ELMo) by using two-layer bidirectional LSTM [9]. More recently, sentence or document encoders that produce contextual token representations have been processed by two steps: pre-trained from unlabeled text and fine-tuned for a supervised downstream task. These approaches can decrease the number of parameters to learn from scratch. One such attempt is Generative Pre-Training (GPT-2) which enhances the context-sensitive embedding [20]. It achieved previously state-of-the-art results in many sentence-level tasks including paraphrase identification from General Language Understanding Evaluation (GLUE) benchmark datasets [21]. However, the attempt is based on a left-to-right architecture. Therefore, every token can only attend to previous tokens, which may cause an issue when we apply it to token-level downstream tasks such as question answering and sentiment analysis.

Devlin et al. focused on the problem and presented a method, BERT, to pre-train deep bidirectional representations from an unlabeled text by jointly conditioning on both left and right context in all layers [11]. They adopted a Masked Language Model (MLM) by adding a next sentence prediction task into the pre-training to learn text-pair representations and can pre-train a deep bidirectional Transformer. Since then, BERT has realized a breakthrough in sentence representation learning which is broadly applied to various NLP tasks including the paraphrase identification task. Lample et al. extended the pre-training model to multiple languages and showed the effectiveness of cross-lingual pre-training. They attempted to integrate two approaches to learn cross-lingual language models (XLMs): the two unsupervised methods, i.e., Causal Language Modeling (CLM) and Masked Language Modeling (MLM), and a supervised method [22]. CLM consists of a Transformer Language model while MLM is based on the technique of Devlin et al. [11]. The supervised model, translation language modeling (TLM) is to improve cross-lingual pre-training which is based on MLM. The common framework related to pre-training mentioned in the above utilizes the Transformer that is the first full-attentional mechanism for learning long-term dependency [23]. Moreover, several approaches apply pre-trained language representation to a large variety of tasks such as named entity, semantic closeness including paraphrase identification and discourse relations through multi-task learning techniques [24–26].

Similar to the recent upsurge of pre-trained contextualized word/sentence representations, graph neural networks [27] such as GCNs [12–14], R-GCNs [28], and Densely Connected GCNs [29] have been successfully employed for many NLP tasks. Such attempts include neural machine translation (NMT) [30,31], pronoun resolution [32], relation extraction [33], semantic role labeling [34] and text classification [35–37]. Most of these attempts showed that the models have contributed to improving the performance on each task, while it has so far not been used for the paraphrase identification task. Moreover, most of them focus on one type of features, syntactic information, and integrate them into their graph model.

3. LSSE Learning Model

Our model leverages various contextual features obtained from the paraphrase-labeled data. Figure 2 illustrates our Lexical, Syntactic, and Sentential Encoding (LSSE) learning framework. The left-hand side of Figure 2 illustrates the overview of our LSSE and the right-hand side is its corresponding flow of the input/output.

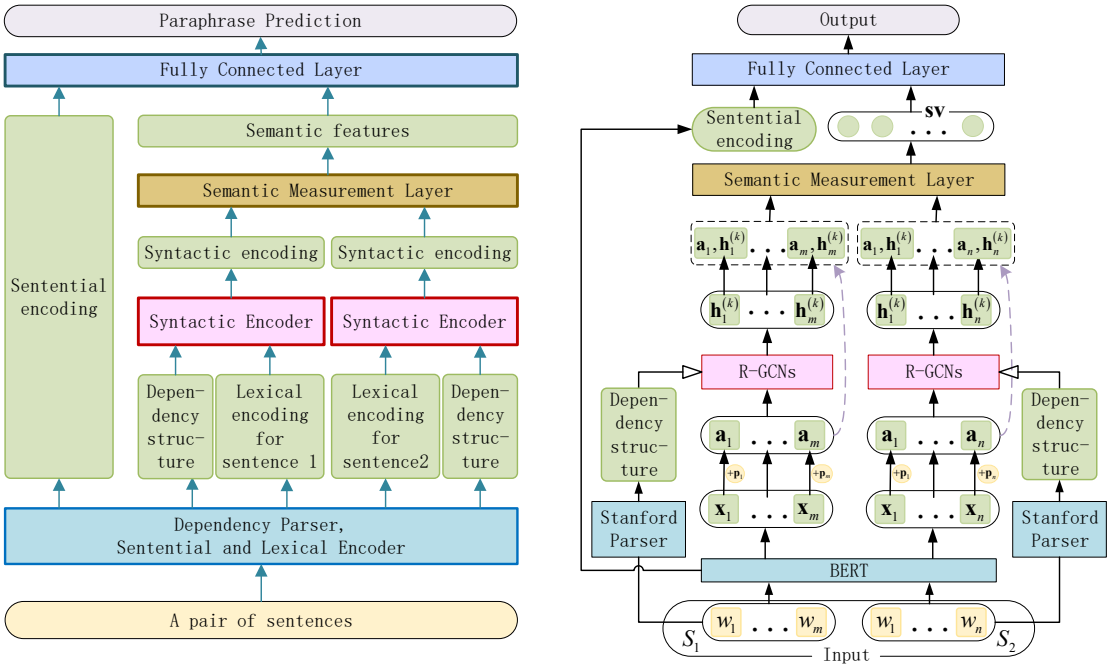

Figure 2. LSSE learning model: S_1 consisting a sequence of $w_1, w_2, \cdots, w_m$ and S_2 which consists of $w_1, w_2, \cdots, w_n$ are a pair of the input sentences. $\mathbf{x}_i$ refers to a word encoding and $\mathbf{p}_i$ indicates its position encoding. $\mathbf{a}_i \in \mathbb{R}^d$ refers to the encoded node feature, i.e., it is obtained by summing up word and position encodings. $\mathbf{h}_i^{(k)} \in \mathbb{R}^d$ is the hidden state of node v_i (w_i) in the $k+1$-th layer.

3.1. Lexical and Sentential Contexts Learning with BERT

The contextualized word representation that we use is BERT which is a Bidirectional Transformer model [11]. A transformer encoder computes the representation of each token through an attention mechanism concerning the surrounding tokens.

BERT architecture consists of two steps: pre-training and fine-tuning. The pre-training BERT model is trained on unlabeled data over different pre-training tasks. It can be easily fine-tuned for NLP tasks by just adding a fully-connected layer. It is pre-trained by using a combination of masked token prediction and next sentence prediction tasks. The input of the BERT is two sentences that are concatenated by a special token [SEP]. It consists of tokens that are segmented by BERT tokenizer using WordPiece embeddings vocabulary [38]. The representation of each token is the sum of the corresponding token, segment, and position embeddings. The first token of every input is the special token of [CLS], and the final hidden state corresponding to this [CLS] token is regarded as an aggregated representation of the input sentence pair. We used this aggregated representation as our sentential encoding of two sentences as well as each token embeddings.

3.2. Syntactic Context Learning with R-GCNs

We utilize R-GCNs to learn syntactic context. It can capture syntactic dependency structures naturally as well as word order because it allows the information to flow in the opposite direction of edges. For example, the sentence in the top of Figure 1, the word "0.2–0.4" modifies the word "percent".

Let S be a sentence and w_i be the i-th absolute position word within the sentence. Let also $G = (V, E)$ be a directed graph, where each node $v_i \in V$ indicates the information of word w_i, consisting of a word encoding $\mathbf{x}_i$. BERT uses word pieces and not word embeddings. When w_i consists of several word pieces, we obtained the average value of all pieces corresponding to w_i and set it to the w_i embeddings. and its position encoding $\mathbf{p}_i$ shown in Figure 2. We can define a matrix $\mathbf{A} \in \mathbb{R}^{d \times n}$ where each column $\mathbf{a}_i \in \mathbb{R}^d$ refers to the encoded node feature of v_i, i.e., we sum up word and position encodings as lexical encoding, $\mathbf{a}_i = \mathbf{x}_i + \mathbf{p}_i$. An edge from node v_i to v_j with a dependency relation type (label) $l \in L$ is denoted by $<v_i, v_j, l> \in E$, where L is a set of dependency relation types. Figure 3 illustrates dependency relations consisting of two information flows: from head to dependent and self-loop. Self-loop is to ensure that the representation of the encoded node feature at the k + 1-th hidden layer can also be informed by its corresponding representation at the k-th hidden layer [28].

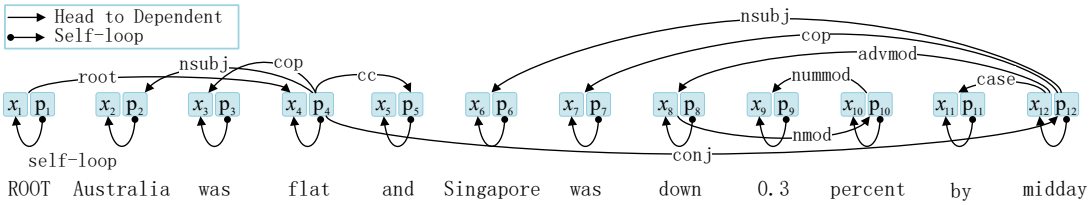

Figure 3. Dependency relations: "$\mathbf{x}_i$" and "$\mathbf{p}_i$" refer to the i-th word encoding and its position encoding, respectively. Arcs indicate two types of edges: (i) head to dependent with labeled syntactic relations such as *nsubj*(*nominal subject*) and *nummod* (*numeric modifier*); and (ii) self-loop.

The propagation model for calculating the forward-pass update of a node v_i in a local graph can be given by:

$$\mathbf{h}_i^{(k+1)} = f\left[\sum_{l \in L}\sum_{j \in N_i^l}\left(\frac{1}{c_i^l}\mathbf{W}_l^{(k)}\mathbf{h}_j^{(k)} + \mathbf{b}_l^{(k)}\right) + \mathbf{W}_0^{(k)}\mathbf{h}_i^{(k)}\right], \quad (1)$$

where $\mathbf{h}_i^{(k)} \in \mathbb{R}^d$ is the hidden state of node v_i in the k-th layer of the neural network with d being the dimensionality of the hidden representations, especially the initial value of $\mathbf{h}_i^{(0)}$ equals to $\mathbf{a}_i$. N_i^l refers to the set of neighbor indices of node v_i under dependency label $l \in L$. c_i^l shows a normalization constant [28]. It can either be learned or chosen in advance. We empirically set c_i^l to 2 in the experiments. $\mathbf{W}_l^{(k)} \in \mathbb{R}^{d \times d}$ stands for the weight matrix and $\mathbf{b}_l^{(k)} \in \mathbb{R}^d$ refers to the bias vector under label $l \in L$ of the k-th hidden layer. We used 32 syntactic dependency relation types including *nsubj* and *dobj* provided by the Stanford parser for the first type of flows and their opposite direction types which would result in having 64 (32 × 2) dependency labels. $\mathbf{W}_0^{(k)} \in \mathbb{R}^{d \times d}$ indicates self-loop convolution weights and f refers to an activation function. We use the ReLU function. Equation (1) shows that it accumulates transformed feature vectors of neighboring nodes which depend on the relation type and the flow of an edge through a normalized sum. Motivated by the method of Vashishth et al. [39], we also utilized a special gate mechanism. Our context learning model is given by:

$$\mathbf{h}_i^{(k+1)} = f\left[\sum_{l \in L}\sum_{j \in N_i^l} g_{lj}^{(k)} \cdot \left(\frac{1}{c_i^l}\mathbf{W}_l^{(k)}\mathbf{h}_j^{(k)} + \mathbf{b}_l^{(k)}\right) + \mathbf{W}_0^{(k)}\mathbf{h}_i^{(k)}\right], \quad (2)$$

where $g_{lj}^{(k)}$ is given by:

$$g_{lj}^{(k)} = \sigma\left(\hat{\mathbf{W}}_l^{(k)}\mathbf{h}_j^{(k)} + \hat{\mathbf{b}}_l^{(k)}\right). \quad (3)$$

$g_{lj}^{(k)}$ is the so-called gate mechanism [34,40] which is to reduce the effect of false dependency edges. The information from neighboring nodes may not be reliable as the dependency relations obtained by some NLP tools are not perfect. Therefore, it needs to be down-weighted. Similar to [32,34], we use the gate value obtained by Equation (3). σ refers to the sigmoid function so that the gate value ranging from 0 to 1. $\hat{\mathbf{W}}_l^{(k)} \in \mathbb{R}^{d\times d}$ and $\hat{\mathbf{b}}_l^{(k)} \in \mathbb{R}^d$ show weights and a bias for the gate under label $l \in L$ of the k-th hidden layer, respectively.

Figure 4 illustrates the R-GCNs model. The left-hand side of Figure 4 is the flow of the model and the right-hand side shows Graph Convolution in the R-GCNs. In the Graph Convolution part shown in the right-hand side of Figure 4, the update of a single node marked with red is computed. Activations from neighboring nodes marked with blue are collected and transformed for each dependency relation such as dep_1 and dep_N individually (for both "in" and "outgoing" edges). The results marked with green, each of which corresponds to $g_{lj}^{(k)} \cdot \left(\frac{1}{c_i^l}\mathbf{W}_l^{(k)}\mathbf{h}_j^{(k)} + \mathbf{b}_l^{(k)}\right)$ or $\mathbf{W}_0^{(k)}\mathbf{h}_i^{(k)}$ in Equation (2), are accumulated and passed through an activation function (ReLU). As shown in the left-hand side of Figure 4, in each hidden layer, the Graph Convolution is applied to update the state of each node of the graph. The output of the R-GCNs is the last hidden layer states. For each sentence, we applied R-GCNs.

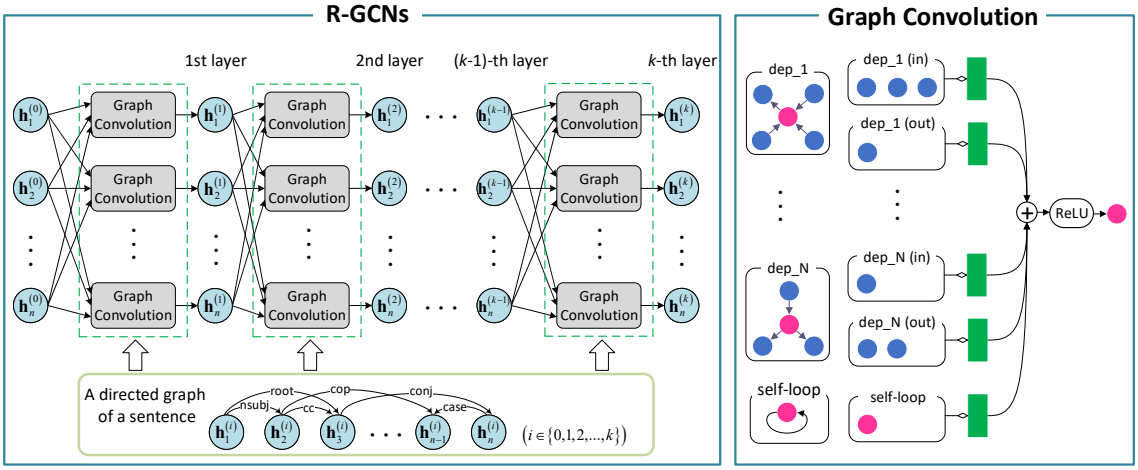

Figure 4. R-GCNs model [28]: The **left**-hand side is the flow of the model and the **right**-hand side shows Graph Convolution in the R-GCNs.

3.3. Paraphrase Identification

Because graph convolution of the R-GCNs model is a special form of Laplacian smoothing, it mixes the features of a node and its neighbors [41,42]. This smoothing operation makes the features of nodes less distinguishable [32]. Therefore, when the training data is small, it often the case that it does not work well. Adding more layers also does not work well as empirically it has been observed that the best performance is achieved with a 2-layer model [41]. Therefore, as illustrated in the right-hand side of Figure 2, after the hidden state $\mathbf{h}_i$ has been learned, we concatenate the hidden state $\mathbf{h}_i$ with the lexical encoding $\mathbf{a}_i$ to keep the original encoding. We obtain the syntactic encoding with the context information aggregated, i.e., $\mathbf{a}'_i = (\mathbf{a}_i, \mathbf{h}_i)$. The result by concatenation has a fixed length, i.e., $2\times d$.

The two matrices $\mathbf{M}_{s_1} \in \mathbb{R}^{2d \times m}$ and $\mathbf{M}_{s_2} \in \mathbb{R}^{2d \times n}$ corresponding to each sentence $S_1 \in \mathbb{R}^{d \times m}$ and $S_2 \in \mathbb{R}^{d \times n}$ are obtained by R-GCNs and passed to the semantic measurement layer which is shown in Figure 5. For each of the two matrices $\mathbf{M}_{s_1}$ and $\mathbf{M}_{s_2}$, we applied the row-based average pooling over them and obtained two vectors, $\mathbf{u}_1$ and $\mathbf{u}_2 \in \mathbb{R}^{2d}$, respectively. We then calculate the similarity between these vectors, i.e., for each dimension, we applied L_1 distance, and obtain a similarity vector $\mathbf{sv} \in \mathbb{R}^{2d}$. The $\mathbf{sv}$ is further concatenated with sentential encoding obtained by BERT, and the result is passed to the fully connected layer **FC**. We set the size of the output layer of the **FC** to two. Finally, we apply the softmax function to obtain probabilities of two predicted labels, paraphrase or non-paraphrase, in the output layer. The network is trained with the objective that minimizes the binary cross-entropy loss of the predicted distributions and the actual distributions (one-hot vectors corresponding to the ground labels) by performing Adam optimization algorithm [43].

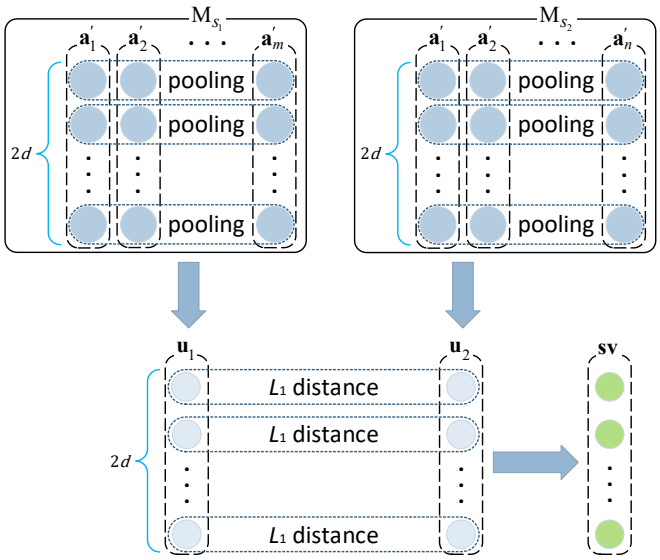

Figure 5. Semantic measurement: after the pooling operation, the similarity between sentences are calculated by using L_1 distance.

4. Experiments

4.1. Experimental Settings

We selected two benchmark datasets, Microsoft Research Paraphrase Corpus (MRPC) [44] and Quora Question Pairs (QQP) [45]. MRPC contains 5801 pairs of sentences extracted from news on the Internet and is annotated to capture the equivalence of paraphrase or semantic relationship between a pair of sentences.

The QQP dataset consists of three folds: 363,870 for training data, 40,431 for validation, and 390,965 for test data. Of these, training and validation data are annotated with a binary classification indicating whether these two questions are a paraphrase or not. We followed the method of Wang et al. [46]. More precisely, we merged training and validation data, and randomly selected 5000 paraphrases and 5000 non-paraphrases as the test set. Table 1 illustrates some sentence pairs from MRPC and QQP. Each data in Table 1 consists of the number of ID, two sentences and its ground labels that the sentences are a paraphrase (1) or non-paraphrase (0).

Table 1. Sentence pairs from MRPC and QQP datasets: Label indicates the ground-truth labels that the sentences are paraphrase (1) or non-paraphrase (0).

Data	#1 ID	#2 ID	#1 String	#2 String	Label
MRPC	2108705	2108831	Yucaipa owned Dominick's before selling the chain to Safeway in 1998 for USD 2.5 billion.	Yucaipa bought Dominick's in 1995 for USD 693 million and sold it Safeway for USD 1.8 billion in 1998.	0
	702876	702977	Amrozi accused his brother, whom he called "the witness", of deliberately distorting his evidence.	Referring to him as only "the witness", Amrozi accused his brother of deliberately distorting his evidence.	1
QQP	364011	490273	What causes stool color to change to yellow?	What can cause stool to come out as little balls?	0
	536040	536041	How do I control my horny emotions?	How do you control your horniness?	1

The paraphrase identification task is a binary classification. Given a pair of sentences, classify them as paraphrases or not paraphrases. All the datasets are parsed by using Stanford parser nlp.stanford.edu/software/lex-parser.shtml [16]. We utilized the BERT_base model as a pre-training model of the lexical and sentential encodings [11] due to the environment with the restricted computational resources. The experiments were conducted on Nvidia TITAN RTX (24GB memory). We used the same model settings as BERT, i.e., the number of training epochs was 3, the batch size was 8, and the number of dimensions of a word and position encoding vectors was 768. The learning rate was 2×10^{-5} by using Adam, learning rate warmup over the first 10,000 steps, and linear decay of the learning rate. We used a dropout probability of 0.1 on all layers in BERT. The number of hidden layers of R-GCNs was optimized by using Optuna https://github.com/pfnet/optuna where the range was [1, 2, 3, 4, 5, 6]. We used 10-fold cross-validation on training data as Phang et al. pointed out that BERT performances become unstable when a training dataset with fine-tuning is small [47]. As a result, we set the number of hidden layers to 2 in the experiments. Following by General Language Understanding Evaluation (GLUE) platform [21], gluebenchmark.com/tasks we used the Accuracy and/or F1-score for evaluation metrics. Throughout the experiments using two benchmark datasets, we choose BERT sentential encodings as a baseline model and implemented a fine-tuning approach in the same manner as with BERT [11].

4.2. Main Results

Table 2 shows the results by using MRPC data (Supplementary Materials). We can see from Table 2 that our model outperformed the baseline, BERT sentential encodings, by 2.0% accuracy and 1.8% F1 on the MRPC and 1.9% accuracy and 2.0% F1 on the QQP data. Why did our LSSE perform particularly strong on the dataset QQP? We notice that the volume of this dataset is larger than that of the MRPC dataset. This confirms our intuition that deep learning typically requires more training data to achieve high performance, and our model could successfully take this advantage on the QQP dataset.

Table 2. Main result by using test dataset: Baseline shows the result obtained by BERT sentential encodings [11]. Bold font shows the best result in each dataset.

Model	Baseline		LSSE	
	Acc	F1	Acc	F1
MRPC	84.3	88.1	**86.3**	**89.9**
QQP	88.7	88.4	**90.6**	**90.4**

Table 3 shows some examples obtained by both of the models. In Table 3, TP, FP, TN, and FN refer to an abbreviation of true positive, false positive, true negative, and false negative, respectively. "N" indicates the number of instances from the test data. For example "N = 70" in Table 3 shows that the number of "LSSE(TP) and BERT(FN)", i.e., the sentence pairs that were classified by LSSE as true positive and classified by BERT as false negative is 70. We can see that the number of "LSSE(TP) and BERT(FN)" is larger than that of "LSSE(FN) and BERT(TP)" in both datasets. However, the number of "LSSE(FP) and BERT(TN)" is larger than that of "LSSE(TN) and BERT(FP)". Most of the errors of FP in our model are in the case that two sentences share the same contents but one sentence has more detailed information of the other. For example in the MRPC dataset, one sentence (#1 String) includes additional information, "private creditors", while it is not mentioned in the second sentence (#2 String). BERT sentential encodings is a simple paraphrase identification compared to our model. But why such a relatively simple model leads to a better prediction for particular test data is not clear at this point. Answering this question requires future research.

Table 3. Example sentences obtained by our LSSE and BERT model: TP, FP, TN and FN refer to an abbreviation of true positive, false positive, true negative and false negative, respectively.

MRPC Dataset				
#1 String	**#2 String**	**LSSE**	**BERT**	**N**
Licensing revenue slid 21 percent, however, to USD 107.6 million.	License sales, a key measure of demand, fell 21 percent to USD 107.6 million.	TP	FN	70
For the entire season, the average five-day forecast track error was 259 miles, Franklin said.	The average track error for the five-day (forecast) is 323 nautical miles.	FN	TP	19
By Sunday night, the fires had blackened 277,000 acres, hundreds of miles apart.	Major fires had burned 264,000 acres by early last night.	TN	FP	36
Other countries and **private creditors** are owed at least USD 80 billion in addition.	Other countries are owed at least USD US80 billion (USD 108.52 billion).	FP	TN	53
QQP Dataset				
#1 String	**#2 String**	**LSSE**	**BERT**	**N**
What are the most intellectually stimulating movies you have ever seen?	What are the most intellectually stimulating films you have ever watched?	TP	FN	331
How do I get business ideas?	How can I think of a business idea?	FN	TP	212
How do I remove dry paint from my clothes?	How do I get acrylic paint out of my clothes?	TN	FP	201
How do Champcash make money from Chrome?	How do a Champcash customer make money from Chrome?	FP	TN	130

We also examined how the percentage of training data affects overall performance. Figure 6 shows an F1-score against the percentage of the MRPC training data. We run ten times for each volume of training data size except for 100% and obtained the average F1-score. Overall, the curves show that more training data helps the performance, while the curves obtained by LSSE drop slowly compared to the BERT sentential encodings. From the observation, we can conclude that our model works well compared to BERT sentential encodings.

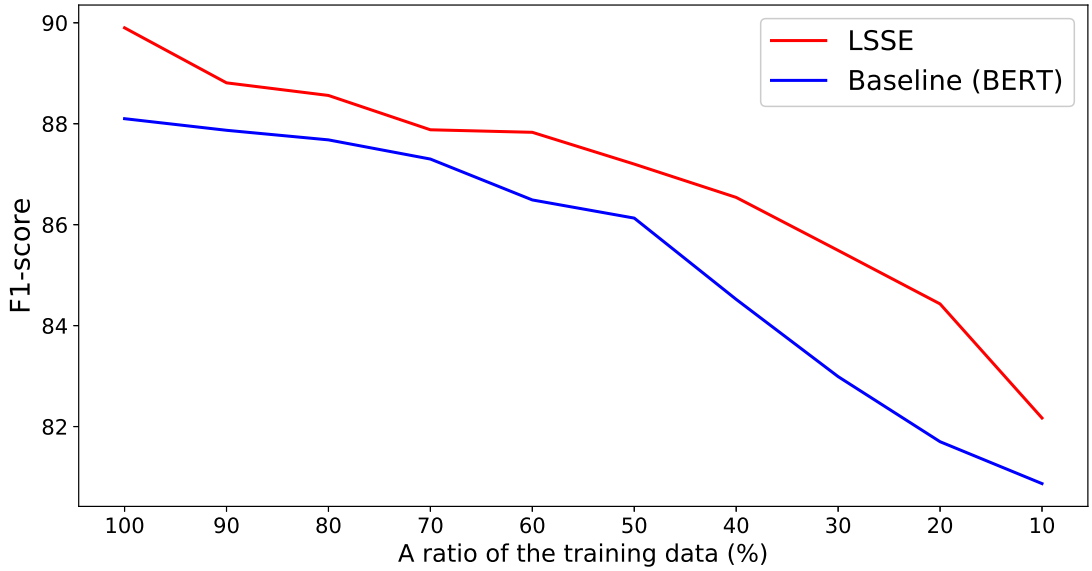

Figure 6. Performance against the percentage of the training data.

4.3. Comparison with Related Work

In MRPC dataset, we compared our model with eight related work, graph-based approach or approaches which utilize BERT_base model to make a fair comparison. These are classified into five types: (i) a relational graph-based approach, Str Align; (ii) BERT_base; (iii) Multi-task learning, GenSen and ERNIE 2.0; (iv) extending BERT pre-trained model, Trans FT, and StructBERT_base; and (v) an adversarial training algorithm, FreeLB-BERT, and its similar approach, ELECTRA.

1. **Str Align**

   Structural Alignment (Str Align) uses a hybrid representation, attributed relational graphs to encode lexical, syntactic and semantic information [48]. To create a relational graph, they used token, lemma, Part-of-Speech (POS) tag, Named Entity Recognition (NER) tag, and Word2Vec word embedding as an attribute of a node, and the dependency label by Stanford CoreNLP is attached to the edge as an attribute. Given two attributed relational graphs, the structural aligner generates an alignment. Then, the similarity score between the two graphs is applied to judge whether they are equivalent or not.

2. **BERT_base model**

   BERT is pre-train deep bidirectional representations from the unlabeled text by jointly conditioning on both left and right context in all layers [11]. We used BERT_base model which contains 12-layers, 12 self-attention heads and 768-dimensional of hidden size.

3. **GenSen**

   GenSen is multi-task learning for sentence representations where a single recurrent sentence encoder is shared across multiple tasks, i.e., multi-lingual NMT, natural language inference, constituency parsing, and skip-thought vectors [49]. The model for multi-task learning is a sequence-to-sequence model. We compared GenSen which utilizes BERT_base model.

4. **ERNIE 2.0**

   Enhanced Representation through kNowledge IntEgration (ERNIE) 2.0 is a multi-task learning model that learns pre-training tasks incrementally [25]. The architecture consists of pre-training and fine-tuning that is the same manner as BERT models. In the process of pre-training, ERNIE 2.0 continually construct unsupervised pre-training tasks with big data and prior knowledge involved, and then incrementally update the model through multi-task learning. In the fine-tuning with task-specific supervised data, the pre-trained model is applied to ten

different NLP tasks in English and nine tasks in Chinese. We compared our model with ERNIE 2.0 using BERT_base model.

5. **Trans FT**

   Transfer Fine-Tuning (Trans FT) is an extended model of BERT to handle phrasal paraphrase relations. The model can generate suitable representations for semantic equivalence assessment instead of increasing the model size [50]. The authors inject semantic relations between a sentence pair into a pre-trained BERT model through the classification of phrasal and sentential paraphrases. After the training, the model can be fine-tuned in the same manner as BERT models. The model achieves improvement on downstream tasks that only have small amounts of training datasets for fine-tuning.

6. **StructBERT_base**

   StructBERT_base incorporates language structures into pre-training BERT_base model [51]. The architecture uses a multi-layer bidirectional Transformer network. It amplifies the ability of the masked language model task by shuffling a certain number of tokens after token masking and predicting the right order. To capture the relationship between sentences, StructBERT randomly swaps the sentence order and predicts the next sentence and the previous sentence as a new sentence prediction task. The model learns the inter-sentence structure in a bidirectional manner as well as to capture the fine-grained word structure in every sentence. In the fine-tuning process, the pre-trained model is applied to a wide range of downstream tasks including GLUE benchmark, Stanford Natural Language inferences (SNLI corpus) and extractive question answering (SQuADv1.1) with good performance.

7. **FreeLB-BERT**

   Free-Large-Batch aims to improve the generalization of pre-trained language models such as BERT, RoBERTAa [52], ALBERT [53] and T5 [54] by enhancing their robustness in the embedding space during finetuning on the downstream language understanding tasks [55]. The method adds norm-bounded adversarial perturbations to the embeddings on the input sentences by using a gradient-based method. Their technique on embedding-based adversaries can manipulate word embeddings which makes it produce powerful pre-trained language models. The results achieved new state-of-the-art on GLUE and AI2 Reasoning Challenge (ARC) benchmark datasets.

8. **ELECTRA-Base**

   "Efficiently Learning an Encoder that Classifies Token Replacements Accurately" (ELECTRA) pre-trains the network as a discriminator that predicts for every token whether it is an original or a replacement. The model trains two neural networks, a generator, and a discriminator. For a given position, the discriminator predicts whether the token of this position comes from the data rather than the generator distribution. The generator is trained to perform masked language modeling. After pre-training, the model fine-tune the discriminator on downstream tasks. ELECTRA-Base that we compared it with our LSSE model is pre-trained in the same manner as BERT_base model.

The results are shown in Table 4 (Supplementary Materials). We can see from Table 4 that LSSE showed a 1.5% accuracy and 1.0% F1-score improvement over BERT_base model. Moreover, our model is competitive for the best systems except for ELECTRA_Base, as ELECTRA_Base outperformed our LSSE by 0.3% in accuracy. This shows that our model can leverage contextual features obtained from the limited volume of the paraphrase-labeled data. We also compared our model with two approaches by using the QQP dataset.

Table 4. Comparative results with related work including state-of-the-art method: Str Align is based on attributed relational graphs. Bold font shows the best result.

MRPC Dataset		
Model	**Acc**	**F1**
Str Align [48]	78.3	84.9
BERT_base [11]	84.8	88.9
GenSen [49]	78.6	84.4
ERNIE 2.0 [25]	86.1	**89.9**
Trans FT [50]	-	89.2
StructBERT_base [51]	86.1	**89.9**
FreeLB-BERT [55]	83.5	88.1
ELECTRA-Base [56]	**86.7**	-
LSSE (Our model)	86.3	**89.9**

1. **BiMPM**

   A Bilateral Multi-Perspective Matching (BiMPM) model [46] encodes given two sentences with a BiLSTM encoder and the two encoded sentences are matched two directions. In each matching direction, each time step of one sentence is matched against all time-steps of another sentence from multiple perspectives. Then, another BiLSTM layer is utilized to aggregate the matching results into a fixed-length matching vector. Finally, a decision is made through a fully connected layer. The authors reported that the experimental results on standard benchmark datasets including QQP showed that the model achieved state-of-the-art performance on all the tasks.

2. **SSE**

   Shortcut-Stacked Sentence Encoder Model (SSE) is a model which enhances multi-layer BiLSTM with skip connection to avoid training error accumulation [57,58]. The input of the k-th BiLSTM layer which is the combination of outputs from all previous layers represents the hidden state of that layer in both directions. The final sentence embedding is the row-based max pooling over the output of the last BiLSTM layer. The experimental results by using eight benchmark datasets including QQP dataset shows that SSE improvs overall performance compared with the three baselines, InferSent [59], Pairwise word interaction model [60], and the decomposable attention model [61], especially it works well in the case that the number of training data is small.

Table 5 shows the results (Supplementary Materials). As we can be seen clearly from Table 5, LSSE outperforms two baseline models as the improvement is 2.4∼2.8%. This indicates that our model works well compared with the sequence model and sentence encoding model based on BiLSTM.

Table 5. Comparative restuls in accuracy by using QQP: Bold font shows the best result.

QQP Dataset	
Model	**Acc**
BiMPM [46]	88.2
SSE [58]	87.8
LSSE (Our model)	**90.6**

4.4. Ablation Study

We recall that our model utilizes lexical and syntactic encodings including the baseline model. Moreover, the syntactic encoding integrates different features. We thus conducted ablation studies to empirically examine the impact of these features/encodings. The results are shown in Table 6.

Table 6. Ablation test: "PE" refers to position encoding and "SentE" indicates sentential encoding. "BERT tokenE" stands for lexical encoding by BERT. "–X" indicates the result by using LSSE without "X". Bold font shows the best result.

MRPC Dataset		
Model	**Acc**	**F**
LSSE (Our model)	**86.3**	**89.9**
–PE	85.3	89.0
–SentE	84.2	88.3
–SentE and –PE	83.5	88.1
–R-GCNs	85.7	89.5
–R-GCNs and –SentE	83.0	87.3
–R-GCNs and –BERT tokenE	84.3	88.1

Table 6 shows the results by using the MRPC dataset (Supplementary Materials). Overall, we can see that integrating different features from the contexts is effective as LSSE was the best performance. The results both without R-GCNs and BERT token encoding (–R-GCNs and –BERT TokenE) and without R-GCNs and sentential encoding (–R-GCNs and –SentE) are worse than those without R-GCNs (-R-GCNs). This shows that the combination of the sentential and lexical encoding is effective for paraphrase identification.

We note that the result by "–SentE" is better than that with "–SentE and –PE". This means that the combination of R-GCNs output, BERT token encoding and position encoding is better than that with only R-GCNs output and BERT token encoding. We can see a similar observation that the combination of sentential encoding, R-GCNs output, BERT token encoding, and position encoding more works well than that with sentential encoding, R-GCNs output, and BERT token encoding because our LSSE is better than the result by "–PE". From these observations, we can conclude that the combination of syntactic features and position encoding contributes to performance improvement.

4.5. Qualitative Analysis of Errors

We performed an error analysis by using the MRPC dataset to provide feedback for further improvement of our method. The number of false-positive and false-negative pairs of sentences was 61 and 38, respectively. These errors have occurred even though we used all the features or any combination of these features. We found that there are mainly three types of errors.

1. **Inclusion relation between sentences:** As we mentioned in Table 3, this error is that two sentences share the same contents but one sentence has more detailed information of the other.

   (1) "There's a Jeep in my parents' yard right now that's not theirs", said Perry, whose parents are vacationing in North Carolina.
   (2) "There's a Jeep in my parents' yard right now that's not theirs", she said.

   Sentence (1) and (2) are similar content and our model identified these sentences as paraphrases. However, according to the Microsoft Research definitions, https://www.microsoft.com/en-us/download/details.aspx?id=52398 these sentences should be identified as "non-paraphrase" because the sentence (1) includes the information marked with the underlined that "Perry's parents are vacationing in North Carolina" and it is a significantly larger superset of the sentence (2). We observed that 39 pairs were classified into this type.
2. **Dependency relation:** Dependency relation within a sentence is not correctly analyzed. For example, in the sentence (3), "<.DDJ>" is divided into four tokens("<", ".", "DDJ", and ">") by BERT tokenizer. As a result, the Stanford parser incorrectly analyzed that ">" modifies "added" with adverb modifier (advmod) relation. In total, 10 pairs of sentences were classified into this type.

(3) The Dow Jones industrial average <.DJI> added 28 points, or 0.27 percent, at 10,557, hitting its highest level in 21 months.

3. **Inter-sentential relations:** Two sentences which have inter-sentential relations are difficult to interpret correctly whether these sentences are paraphrase or not.

(4) British Airways' New York-to-London runs will end in October.
(5) British Airways plans to retire its seven Concordes at the end of October.

Sentences (4) and (5) have the same sense, while different expressions such as "New York-to-London" and "Concordes" are used and they are co-referred entities. To identify these sentences as "paraphrases" correctly, it requires not only local dependency, i.e., dependency structure within a sentence but also non-local dependency between sentences. There were nine pairs classified into this type.

Apart from these observations, we found that when the number of arcs from other nodes is small, the performance of R-GCNs has not improved because convolution mixes the features of a node and its neighbors. One solution is to incorporate more linguistics information such as tree-based structure [62,63], Named Entity Recognition, and Co-Reference Resolution into our framework to represent rich relations among nodes. This is a rich space for further exploration.

We recall that our model for lexical and sentential encodings are based on the BERT. The BERT pre-training model, an unsupervised manner is to learn general, domain-independent knowledge. However, most of the downstream tasks including paraphrase identification and even in the same task, there are several domain-specific data which are collected from different genres such as MRPC and QQP. It would be helpful to develop a good fine-tuning method in our future work.

5. Conclusions

We focused on the problem that how to interpret a diversity of context information as relevant features and proposed an approach by leveraging a variety of features with a neural-based learning model. For syntactic encodings, our LSSE model incorporates word encoding, position encoding, and full dependency structures into a unified framework, R-GCNs. By utilizing the hidden states obtained by the R-GCNs as well as lexical and sentential encodings by BERT, our model learns contextual similarity between sentences. The experimental results by using two datasets showed that our model attained at 86.3% accuracy and 89.9% F1-score in MRPC, and 90.6% accuracy in QQP data which are comparable to the related work on paraphrase identification methods. Moreover, throughout the ablation test, we found that the combination of position encoding and syntactic features contributes to performance improvement.

There are several interesting directions for future work. We should be able to obtain further advantages in efficacy in our syntactic embeddings obtained by the R-GCNs model. We empirically examined that the best performance is achieved with a two-layer model, while R-GCNs with more layers can be considered to capture richer neighborhood information of a graph. Guo et al. focused on this problem and proposed a densely connected graph convolutional network that introduces residual connections, dense connectivity, and graph attention techniques [29]. They reported that the model attained at the current state-of-the-art neural models in the English–German and English–Czech translation tasks. This is definitely worth trying with our LSSE learning model.

As we mentioned in Section 4.5, we found that more effective knowledge extraction improves the overall performance of paraphrase identification. Our model utilized BERT_base model for lexical and sentential encodings and applied it to two domain-specific data, MRPC and QQP. However, the BERT pre-training model is to learn general domain-independent knowledge. In the phase of fine-tuning, the model learns by using these domain-specific data which causes difficulty to estimate optimal parameters. Moreover, Phang et al. reported that BERT is unstable when a training dataset

with fine-tuning is small [47]. One approach is to develop a knowledge transfer technique which is some empirical work along these lines in the deep learning field [64]. This is a rich space for further exploration.

Supplementary Materials: The following are available online at http://www.mdpi.com/2076-3417/10/12/4144/s1. Tables 2, 4, 5, 6.

Author Contributions: Conceptualization, F.F., and S.X.; methodology, S.X. and F.F.; software, S.X.; validation, S.X., F.F., X.S., and J.L.; investigation, S.X. and F.F.; writing—original draft preparation, S.X.; writing—review and editing, F.F., X.S., J.L., Y.S., and H.N.; supervision, F.F., and X.S.; funding acquisition, F.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Grant-in-aid for JSPS, Grant Number 17K00299, and Support Center for Advanced Telecommunications Technology Research, Foundation.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. He, H.; Gimpel, K.; Lin, J. Multi-perspective Sentence Similarity Modeling with Convolutional Neural Networks. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 1576–1586.
2. Yin, W.; Schütze, H.; Xiang, B.; Zhou, B. ABCNN: Attention-based Convolutional Neural Network for Modeling Sentence Pairs. *Trans. Assoc. Comput. Linguist.* **2016**, *4*, 259–272. [CrossRef]
3. Liu, P.; Qiu, X.; Huang, X. Modelling Interaction of Sentence Pair with Coupled-LSTMs. *arXiv* **2016**, arXiv:1605.05573.
4. Chen, Q.; Zhu, X.; Ling, Z.; Wei, S.; Jiang, H.; Inkpen, D. Enhanced LSTM for Natural Language Inference. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1657–1668.
5. Wieting, J.; Gimpel, K.Revisiting Recurrent Networks for Paraphrastic Sentence Embeddings. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics,Vancouver, BC, Canada, 30 July–4 August 2017; pp. 2078–2088.
6. Wang, Y.; Huang, H.; Chong, F.; Zhou, Q.; Jiahui, G.; Xiong, G. CSE: Conceptual Sentence Embeddings based on Attention Model. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016; pp. 505–515.
7. Oren, M.; Jacob, G.; Ido, D. Context2vec: Learning Generic Context Embedding with Bidirectional LSTM. In Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, Berlin, Germany, 11–12 August 2016; pp. 51–61.
8. Sanjeev, A.; Yingyu, L.; Tengyu, M. A Simple but Tough-to-Beat Baseline for Sentence Embeddings. In Proceedings of the 5th International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
9. Mark, N.; Mohit, I.; Matt, G.; Christopher, C.; Kenton, L.; Luke, Z. Deep Contextualized Word Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA, 1–6 June 2018; pp. 2227–2237.
10. Cer, D.; Yang, Y.; Kong, S.; Hua, N.; Limtiaco, N.; St. John, R.; Constant, N.; Guajardo-Cespedes, M.; Yuan, S.; Tar, C.; et al. Univeral Sentence Encoder. *arXiv* **2018**, arXiv:1803.11175.
11. Jacob, D.; Ming-Wei, C.; Kenton, L.; Kristina, T. BERT: Pre-training on Deep Bidirectional Transfomers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; pp. 24171–4186.
12. Michaël, D.; Xavier, B.; Pierre, V. Convolutiona Neural Networks on Graphs with Fast Localized Sectral Filtering. In Proceedings of the 30th Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 3844–3852.
13. Thomas, K.; Max, W. SEMI-Supervised Classification with Graph Convolutional Networks. In Proceedings of the 5th International Conference on Learning Representations, Toulon, France, 24–26 April 2017.

14. Felix, W.; Tianyi, Z.; de, S.J.A.H.; Christopher, F.; Tao, Y.; Q, W.K. Simplifying Graph Convolutional Networks. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6861–6871.
15. Socher, R.; Bauer, J.; Manning, C.D.; Ng, A.Y. Parsing with Compositional Vector Grammars. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; pp. 455–465.
16. Danqi, C.; Manning, C.D. A Fast and Accurate Dependency Parser using Neural Networks. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 740–750.
17. Socher, R.; Huang, E.H.; Pennington, J.; Ng, A.Y.; Manning, C.D. Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection. In *Advances in Neural Information Processing Systems*; The MIT press: Cambridge, MA, USA, 2011; pp. 801–809.
18. Hu, B.; Lu, Z.; Li, H.; Chen, Q. Convolutional Neural Network Architectures for Matching Natural Language Sentences. In *Advances in Neural Information Processing Systems*; The MIT press: Cambridge, MA, USA, 2015; pp. 2042–2050.
19. Tai, K.S.; Socher, R.; Manning, C.D. Improved Semantic Representations from Tree-structured Long Short-term Memory Networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, Beijing, China, 26–31 July 2015; pp. 1556–1566.
20. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
21. Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S.R. GLUE: A Multi-task Benchmark and Analysis Platform for Natural Language Understanding. *arXiv* **2018**, arXiv:1804.07461.
22. Lample, G.; Conneau, A. Cross-lingual Language Model Pretraining. *arXiv* **2019**, arXiv:1901.07291.
23. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All You Need. In *Advances in Neural Information Processing Systems*; The MIT press: Cambridge, MA, USA, 2017; pp. 5998–6008.
24. Subramanian, S.; Trischler, A.; Bengio, Y.; Pal, J.C. Learning General Purpose Distributed Sentence representations via Large Scale Multi-task Learning. In Proceedings of the 6th International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
25. Sun, Y.; Wang, S.; Li, Y.; Feng, S.; Tian, H.; Wu, H.; Wang, H. ERNIE 2.0: A Continual Pre-training Framework for Language Understanding. *arXiv* **2019**, arXiv: 1907.12412.
26. Liu, X.; He, P.; Chen, W.; Gao, J. Multi-task Deep Neural Networks for Natural Language Understanding. *arXiv* **2019**, arXiv:1901.11504.
27. Wu, Z.; Pan, S.; chen, F.; Long, G.; Zhang, C.; Yu, P.S. A Comprehensive Survey on Graph Neural Networks. *arXiv* **2019**, arXiv:1901.00596.
28. Michael, S.; N, K.T.; Peter, B.; Rianne, V.D.B.; Ivan, T.; Max, W. Modeling Relational Data with Graph Convolutional Networks. In Proceedings of the European Semantic Web Conference, Crete, Greece, 3–7 June 2018; pp. 593–607.
29. Zhijiang, G.; Yan, Z.; Zhiyang, T.; Wei, L. Densely Connected Graph Convolutional Networks for Graph-to-Sequence Learning. *Trans. Assoc. Comput. Linguist.* **2019**, *7*, 297–312.
30. Joost, B.; Ivan, T.; Wilker, A.; Diego, M.; Khalil, S. Graph Convolutional Encoders for Syntax-aware Neural Machine Translation. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 1957–1967.
31. Beck, D.; Haffari, G.; Cohn, T. Graph-to-Sequence Learning using Gated Graph Neural Networks. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; pp. 273–283.
32. Yinchuan, X.; Junlin, Y. Look Again at the Syntax: Relational Graph Convolutional Network for Gendered Ambiguous Pronoun Resolution. In Proceedings of the 1st Workshop on Gender Bias in Natural Language Processing, Florence, Italy, 2 August 2019; pp. 99–104.
33. Zhijiang, G.; Yan, Z.; Wei, L. Attention Guided Graph Convolutional Networks for Relation Extraction. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 241–251.

34. Diego, M.; Ivan, T. Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 1506–1515.
35. Hamilton, W.L.; Ying, R.; Leskovec, J. Inductive representation learning on large graphs. In Proceedings of the 31st Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 1024–1034.
36. Peter, V.; Guillem, C.; Arantxa, C.; Adriana, R.; Pietro, L.; Yoshua, B. Graph Attention Networks. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
37. Liang, Y.; Chengsheng, M.; Yuan, L. Graph Convolutional Networks for Text Classification. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 7370–7377.
38. Wu, Y.; Schuster, M.; Chen, Z.; Le, Q.V.; Norouzi, M. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *arXiv* **2016**, arXiv:1609.08144.
39. Shikhar, V.; Manik, B.; Prateek, Y.; Piyush, R.; Chiranjib, B.; Partha, T. Incorporating Syntactic and Semantic Information in Word Embeddings using Graph Convolutional Networks. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 3308–3318.
40. Van den Oord, A.; Kalchbrenner, N.; Espeholt, L.; Vinyals, O.; Graves, A. Conditional Image Generation with PixelCNN Decoders. In Proceedings of the 30th Conference on Neural Information Processin System, Barcelona, Spain, 5–10 December 2016; pp. 4790–4798.
41. Li, Q.; Han, Z.; Wu, X.M. Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. In Proceedings of the 32nd AAAI conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 3538–3545.
42. Yang, L.; Kang, Z.; Can, X.; Jin, D.; Yang, B.; Guo, Y. Topology Optimization based Graph Convolutional Network. In Proceedings of the 28th International Joint Conference on Artificial Intelligence, Macao, China, 10–16 August 2019; pp. 4054–4061.
43. Kingma, D.P.; Ba, J. ADAM: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015; pp. 1–15.
44. Dolan, W.B.; Brockett, C. Automatically Constructing a Corpus of Sentential Paraphrases. In Proceedings of the Third International Workshop on Paraphrasing, Jeju Island, Korea, 14 October 2005; pp. 9–16.
45. Shankar, I.; Nikhil, D.; Kornél, C. First Quora Dataset Release: Question Pairs. 2016. Available online: https://data.quora.com/First-Quora-Dataset-Releasee-Question-Pairs (accessed on 1 March 2020.)
46. Zhiguo, W.; Wael, H.; Radu, F. Bilateral Multi-Perspective Matching for Natural Language Sentences. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 4144–4150.
47. Phang, J.; Fevry, T.; Bowman, S.R. Sentence Encoders on STILTs: Supplementary Training on Intermediate Labeled-data Tasks. *arXiv* **2019**, arXiv: 1811.01088.
48. Liang, C.; Paritosh, P.K.; Rajendran, V.; Forbus, K.D. Learning Paraphrase Identification with Structural Alignment. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, New York, NY, USA, 9–15 July 2016; pp. 2859–2865.
49. Subramanian, S.; Trischler, A.; Bengio, Y.; Pal, C.J. Learning General Purpose Distributed Sentence Representations via Large Scale Multi-task Learning. *arXiv* **2018**, arXiv:1804.00079.
50. Yuki, A.; Junichi, T. Transfer Fine-Tuning: A BERT Case Study. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 3–7 November 2019; pp. 5393–5404.
51. Wei, W.; Bin, B.; Ming, Y.; Chen, W.; Zuyi, B.; Liwei, P.; Luo, S. StructBERT: Incorporating Language Structures into Pre-training for Deep Language Understanding. *arXiv* **2019**, arXiv:1908.04577.
52. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv: 1907.11692.
53. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In Proceedings of the 8th International Conference on Learning Representations, Addis Ababa, Ethiopia, 26 April–1 May 2020.

54. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv* **2019**, arXiv: 1910.10683.
55. Chen, Z.; Yu, C.; Zhe, G.; Siqi, S.; Thomas, G.; Jing, L. FreeLB: Enhanced Adversarial Training for Natural Language Understanding. *arXiv* **2019**, arXiv:1909.11764.
56. Clark, K.; Luong, M.T.; Le, Q.V.; Manning, C.D. ELECTRA: Pre-Training Text Encoders as Discriminators Rather than Generators. In Proceedings of the 8th International Conference on Learning Representations, Addis Ababa, Ethiopia, 26 April–1 May 2020.
57. Nie, Y.; Bansal, M. Shortcut-stacked Sentence Encoders for Multi-Domain Inference. In Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP, Copenhagen, Denmark, September 2017; pp. 41–45.
58. Wuwei, L.; Wei, X. Neural Network Models for Paraphrase Identification Semantic Textual Similarity Natural Language Inference and Question Answering. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; pp. 3890–3902.
59. Conneau, A.; Kiela, D.; Schwenk, H.; Barrault, L.; Bordes, A. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 670–680.
60. He, H.; Lin, J. Pairwise Word Interaction Modeling with Deep Neural Networks for Semantic Similarity Measurement. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 937–948.
61. Parikh, A.; Tom, O.; Das, D.; Uszkoreit, J. A Decomposable Attention Model for Natural Language Inference. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 2249–2255.
62. Moschitti, A. Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees. In Proceedings of the 17th European Conference on Machine Learning and the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases, Berlin, Germany, 18–22 September 2006; pp. 318–329.
63. Moschitti, A.; Chu-Carroll, J.; Patwardhan, S.; Fan, J.; Riccardi, G. Using Syntactic and Semantic Structural Kernels for Classifying Definition Questions in Jeopardy! In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Scotland, UK, 27–31 July 2011; pp. 712–724.
64. Papernot, N.; Abadi, M.; Úlfar, E.; Goodfellow, I.; Talwar, K. Semi-Supervised Knowledge Transfer for Deep Learning from Private Training Data. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.